畅想器的测试分析报告

目录

[一、引言 2](#_Toc389765955)

[1. 编写目的 2](#_Toc389765956)

[2. 背景 2](#_Toc389765957)

[3．参考资料 2](#_Toc389765958)

[4．版本更新信息 2](#_Toc389765959)

[二．测试概要 2](#_Toc389765960)

[1.测试采用的标准及技术 2](#_Toc389765961)

[三、单元测试 3](#_Toc389765962)

[四、集成测试 8](#_Toc389765963)

[五、系统测试 9](#_Toc389765964)

[1.对功能的测试 9](#_Toc389765965)

[2. 对性能的测试 13](#_Toc389765966)

[六、测试结论 14](#_Toc389765967)

## 一、引言

### 1. 编写目的

畅想器是多人联网的工作平台，在项目进行阶段，我们需要进行多项测试，以检测项目存在的问题并对此提出相应的解决方案。

本文档将提供给本项目的开发人员、维护人员查阅和使用。

### 2. 背景

a. 被测试软件名称：畅想器

b. 软件开发人员：孙笑凡、王奥鑫、宋睿、张磊、李小爽、于泽

c. 用户：畅想器可以通过网络实现多人联网自动化办公，多人产品设计，网络会议，网络教学等应用。因此可以满足平面设计者，教师，公司管理人员等工作者的在线工作需求。

d. 测试环境：

(1)操作系统：win7/win8

(2)数据库：MySQL/SQL Server

(3)开发工具：Visual Studio2010/2012/2013

### 3．参考资料

[1]吕云翔 王昕鹏 邱玉龙. 软件工程——理论与实践

### 4．版本更新信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本号 | 创建者 | 创建日期 | 维护者 | 维护日期 | 维护纪要 |
| V1.0 | 孙笑凡 | 2014-4-1 |  |  |  |
|  |  |  |  |  |  |

## 二．测试概要

### 1.测试采用的标准及技术

表1 开始、中断、运行、完成测试

|  |  |
| --- | --- |
| 开始、中断、运行、完成测试 | 标准说明 |
| 开始测试标准 | 硬件运行环境可用且软件能够正确安装完成 |
| 中断测试标准 | 安装无法正确完成或者服务器无法连接或者程序运行终止或异常退出 |
| 运行测试标准 | 程序正常进行，各项功能均使用正常 |
| 完成测试标准 | 完成测试规划，确认能够达到程序和测试质量的目标 |

表2 测试技术

|  |  |
| --- | --- |
| 测试技术 | 说明 |
| 编写测试用例 | 在项目进行编码阶段对各项功能进行测试 |
| 单元测试 | 测试关注单元的具体实现、内部逻辑、数据流向等 |
| 集成测试 | 将所有模块按照概要设计要求组装为子系统或系统进行集成测试 |
| 系统测试 | 将已经集成好的软件系统作为整个计算机系统的一个元素，与计算机硬件、外设、支持软件、数据和人员等其他系统元素结合在一起，在实际运行环境中对计算机系统进行一系列的组装测试和确认测试 |

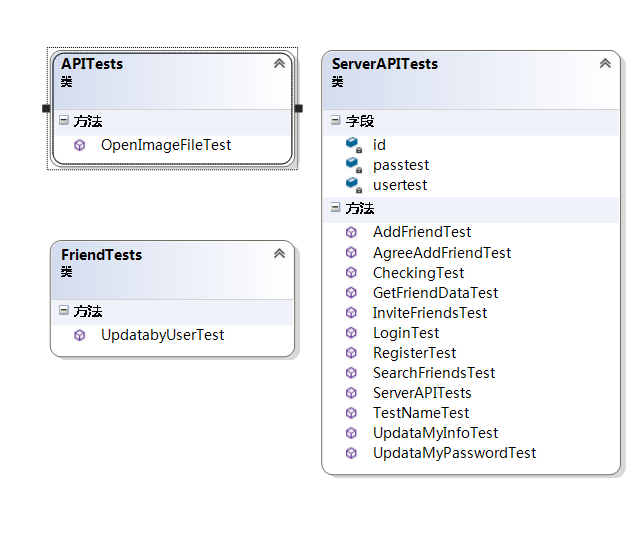
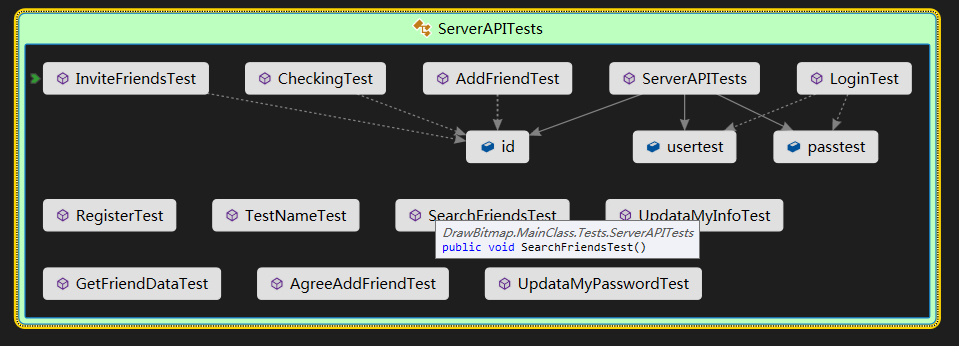
表3 测试类型

|  |  |
| --- | --- |
| 测试类型 | 说明 |
| 功能测试 | 根据需求规格说明书和软件设计说明书所编写的软件功能实现，检查项目是否正确可靠的实现了功能 |
| 边界值测试 | 选择边界数据或非常规数据进行边界值测试，确保项目的稳定性、可靠性，在多数情况下都能保证程序无异常 |
| 界面测试 | 检查界面的美观性以及界面设计是否符合用户体验的需求 |
| 文档测试 | 检查文档编写是否符合要求以及描述是否详细合理 |

## 三、单元测试

由于本工程的代码量比较大，单元测试主要是对类API，ServerAPI，Friend的各种接口，并就结果进行分析。

下图是借助vs针对以上三个类生成对应的测试类



以下是单元测试的细节：

1.ServerAPI测试

ServerAPI测试前，先要进行初始化工作，在立构函数中，设置IP和Port，创建一个服务器

public ServerAPITests()

{

ServerAPI.ServerIP = "111.195.198.161";

ServerAPI.ServerPort = 9999;

ServerAPI.client = new SendingClient(ServerAPI.ServerIP, ServerAPI.ServerPort);

}

2.注册

参数：用户名 密码 昵称

返回：id

注册已经注册的用户应该失败，返回-1

public void RegisterTest()

{

string usertest = "sxf";

string passtest = "12211037";

string nicktest = "sxf";

int expected = -1;//预期的值

int actual;//实际的值

actual = ServerAPI.Register(usertest, passtest, nicktest);

Assert.AreEqual(expected, actual);

}

但其返回了id，我们发现，我们并没有在数据库端进行验证，而是在客户端进行了用户名可用检查。

3.登陆

参数：用户名 密码

返回值是一个登陆时需要的数据结构

public void LoginTest()

{

LoginReturn actual;//实际的值

actual = ServerAPI.Login(usertest, passtest);

Assert.AreNotEqual(null, actual);

}

发现成功返回，不为null

4.确定在线并获取消息

参数：id

返回：List<UserMessage>

public void CheckingTest()

{

List<UserMessage> actual;//实际的值

actual = ServerAPI.Checking(id);

}

发现服务端出现1人在线的消息，成功

5.邀请好友

参数：用户id 插件id 用户id列表

返回：是否成功的邀请每个人

public void InviteFriendsTest()

{

int plugin\_id = 1;

SortedSet<int> id\_list = new SortedSet<int>();

id\_list.Add(44);

ServerAPI.InviteFriends(plugin\_id, id, id\_list);

Assert.AreEqual(true, v);

}

对Set集合内的每一个人都发送一个邀请通知

6.检验名字

参数：待注册的用户名

返回：布尔

public void TestNameTest()

{

int a = ServerAPI.TestName("sxf");

int b = ServerAPI.TestName("www");

Assert.AreEqual(-1, a);

Assert.AreEqual(1, b);

}

如果一个名字已经被注册，则返回-1，没有则返回1

7.查询好友

参数：好友的名字

返回：好友列表

public void SearchFriendsTest()

{

List<User> u = ServerAPI.SearchFriends("wax");

List<User> p = ServerAPI.SearchFriends("www");

Assert.AreEqual(1, u.Count);

Assert.AreEqual(0, p.Count);

}

都会返回一个列表，只不过列表元素个数不同。

如果搜索一个没被注册过的名字，返回的将是一个空集合。

8.添加好友

参数：id1，id2

返回：是否成功将两人添加为好友关系

public void AddFriendTest()

{

int v\_id = 51;

bool v = ServerAPI.AddFriend(v\_id,id);

Assert.AreEqual(false,v);

v\_id = 519;

v = ServerAPI.AddFriend(v\_id, id);

Assert.AreEqual(false, v);

}

尝试添加一个未注册的好友和一个已经成为好友的人会返回false

9.更新用户数据

参数：User类

返回：是否成功执行更新操作

public void UpdataMyInfoTest()

{

User u = new User();

u.name = "wxs";

u.user\_id = 51;

u.nickname = "旭日东升";

u.ext = null;

bool v = ServerAPI.UpdataMyInfo(u);

Assert.AreEqual(true, v);

u.name = "ppp";

u.user\_id = 199;

u.nickname = "sss";

u.ext = null;

v = ServerAPI.UpdataMyInfo(u);

Assert.AreEqual(false, v);

u.name = null;

u.user\_id = 9000;

u.nickname = "sss";

u.ext = null;

v = ServerAPI.UpdataMyInfo(u);

Assert.AreEqual(false, v);

u.name = "ggg";

u.user\_id = 0;

u.nickname = "sss";

u.ext = null;

v = ServerAPI.UpdataMyInfo(u);

Assert.AreEqual(false, v);

}

尝试更新一个不存在的用户将导致问题，不过目前我们的软件并没有通过该测试，这是一个已知bug。

10.更新用户密码

参数：id，旧密码，新密码

返回：是否正确执行更新

public void UpdataMyPasswordTest()

{

int id = 51;

String opass = "ww";

String npass = "www";

int v = ServerAPI.UpdataMyPassword(id,opass,npass);

Assert.AreEqual(1, v);

ServerAPI.UpdataMyPassword(id, opass, npass);

Assert.AreEqual(-1, v);

v = ServerAPI.UpdataMyPassword(id, npass, opass);

Assert.AreEqual(1, v);

}

尝试用一个不正确的旧密码来修改用户密码是做不到的。

11.获取指定用户数据

参数：id

返回：User类

public void GetFriendDataTest()

{

User v = ServerAPI.GetFriendData(20);

Assert.AreEqual("sxf", v.name);

v = ServerAPI.GetFriendData(293);

Assert.AreEqual(null, v);

}

获取存在的人的数据将得到正确的信息，获得不存在的用户，将获得null

12.同意添加用户为好友

参数：id1 id2 是否同意（1为同意）

public void AgreeAddFriendTest()

{

int id\_1 = 51;

int id\_2 = 20;

bool v = ServerAPI.AgreeAddFriend(id\_1,id\_2,1);

Assert.AreEqual(true, v);

v = ServerAPI.AgreeAddFriend(id\_1, id\_2, 1);

Assert.AreEqual(false, v);

v = ServerAPI.AgreeAddFriend(id\_1, id\_2, 0);

Assert.AreEqual(true, v);

}

反复同意添加用户为好友会导致失败，但我们的系统目前并未通过该测试，也是一个已知bug

## 四、集成测试

该项目的集成测试重点是数据流的传输是否正常以及数据流的接收是否正确。

在该项目中，由于有功能好友聊天、多人绘图、文件分享等功能，均涉及到数据流的传输和接收，因此需要对此进行集成测试。

测试主要是进行文本文档、图片、文件夹等各类文件类型的传输和接收。

本系统的集成测试重点是窗口间的跳转是否正确以及窗口间的参数传递是否正确。对于窗口间的跳转测试，主要测试点击本系统中的各个窗口的按钮链接提示是否跳转到相应界面；对于参数传递的测试，主要测试窗口间传递参数所产生的结果是否正确。

表4子窗口显示测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 按钮名称 | 所在的窗口 | 测试结果 | 测试人 | 测试时间 |
| 1 | 添加好友 | 主界面 | 通过 | 王奥鑫 | 2014-5-31 |
| 2 | 修改信息 | 主界面 | 通过 | 王奥鑫 | 2014-5-31 |
| 3 | 设置 | 主界面 | 通过 | 王奥鑫 | 2014-5-31 |
| 4 | 创建群组 | 主界面 | 通过 | 王奥鑫 | 2014-5-31 |
| 5 | 删除联系人 | 主界面 | 通过 | 王奥鑫 | 2014-5-31 |

## 五、系统测试

对于该项目，这里将系统测试分为两个部分：功能测试和性能测试功能测试是系统测试中最基本的测试，验证产品的功能是否符合需求规格，主要包括：功能是否全部实现、功能是否满足用户需求以及能否正确地接受输入，并给出相应正确的结果。对于其中几项主要的功能，我们按照等价类划分的方法进行了测试用例。对于每个测试用例，均以清晰的表格形式记录了测试结果、发现的问题、测试人以及测试时间。

性能测试是用来测试软件系统在实际的集成系统中运行的性能，按照《畅想器软件设计说明书》对性能的要求，设计了性能测试表格。需要的性能级别针对于实际的性能级别进行比较，并把其中的差距文档化。每项测试同功能测试一样均记录了测试结果、发现的问题、测试人及测试时间。

### 1.对功能的测试

表 5 功能测试

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **编号** | **功能名称** | **功能描述** | **输入/需要** | **输出/结果** | **发现问题** | **测试结果** | **测试人** | **测试时间** |
| 1 | 用户注册 | 用于未注册的游客注册成为会员 | 昵称、用户名、密码 | 成功注册或者注册失败 | 无 | 通过 | 张磊 | 2014-5-31 |
| 2 | 用户登录 | 用于已注册的会员登陆系统平台 | 用户名、密码 | 成功登入系统或显示出错信息 | 无 | 通过 | 张磊 | 2014-5-31 |
| 3 | 修改密码 | 用于已注册的会员修改密码 | 用户之前的密码，新密码 | 用户重设密码成功或失败 | 无 | 通过 | 张磊 | 2014-5-31 |
| 4 | 修改个人信息 | 用于已注册的会员修改个人信息 | 用户重新输入个人信息 | 用户的信息修改成功 | 无 | 通过 | 张磊 | 2014-5-31 |
| 5 | 用户注销 | 用于已注册的用户退出系统 | 用户确定注销 | 用户退出系统 | 无 | 通过 | 张磊 | 2014-5-31 |
| 6 | 查找好友 | 用户通过用户名查找好友 | 输入想要添加的好友的用户名 | 成功搜索到相关信息并跳转 | 无 | 通过 | 张磊 | 2014-5-31 |
| 7 | 添加好友 | 用于用户添加好友 | 点击添加按钮 | 添加成功 | 无 | 通过 | 张磊 | 2014-5-31 |
| 8 | 删除好友 | 用于用户删除好友 | 点击删除按钮 | 删除成功 | 无 | 通过 | 张磊 | 2014-5-31 |
| 9 | 发送消息 | 用于用户之间聊天发送消息 | 输入要发送的内容，点击发送按钮 | 消息发送到对方聊天窗口 | 无 | 通过 | 张磊 | 2014-5-31 |
| 10 | 创建群组 | 用于已注册会员创建群组 | 用户选择创建群组 | 创建成功 | 无 | 通过 | 张磊 | 2014-5-31 |
| 11 | 共享文件 | 用于已注册的会员在群组里共享文件 | 用户选择共享文件，选择要共享的文件夹 | 系统提示共享成功，群组中的其他好友可以从中取出文件 | 无 | 通过 | 张磊 | 2014-5-31 |
| 12 | 用画笔绘图 | 用于用户调整想使用的画笔颜色和宽度，在面板上绘出图形 | 用户选择想使用的画笔颜色和宽度，绘制图形 | 选择成功，画笔颜色和宽度改变，画板上出现所绘制的图形 | 无 | 通过 | 张磊 | 2014-5-31 |

下面是对某些严格要求数据正确性的模块的测试设计及测试结果，对于这些模块采用了等价类划分的测试方法。

表6 等价类划分

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入条件 | 有效等价类 | 编号 | 无效等价类 | 编号 |
| 用户名 | 用户名与数据库中一致 | 1 | 用户名与数据库中不一致 | 3 |
| 密码 | 密码与数据库中一致 | 2 | 密码与数据库中不一致 | 4 |

表7 测试用例设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 输入 | | 预期输出 | 覆盖范围 |
| 用户名 | 密码 |
| 1.1 | sxf | 123 | 成功登陆到系统 | 1、2 |
| 1.2 | www | 123 | 登陆失败 | 3 |
| 1.3 | sxf | 654 | 登陆失败 | 4 |

注：我们已经提前注册好了sxf密码为123和www密码为jjj的两个用户。

表8 测试结果记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例编号 | 预期结果 | 实际结果 | 测试结果 | 测试日期 | 测试人 |
| 1.1 | 登陆成功 | 登陆成功 | 通过 | 2014-5-31 | 张磊 |
| 1.2 | 登陆失败 | 登陆失败 | 通过 | 2014-5-31 | 张磊 |
| 1.3 | 登陆失败 | 登陆失败 | 通过 | 2014-5-31 | 张磊 |

表9 等价类划分

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入条件 | 有效等价类 | 编号 | 无效等价类 | 编号 |
| 消息内容 | 长度1-5000之间 | 1 | 长度为0（即未输入） | 2 |
| 长度大于5000 | 3 |

表10 测试用例设计

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 输入 | 预期输出 | 覆盖范围 |
| 输入内容 |
| 2.1 | 123 | 发送成功 | 1 |
| 2.2 | 空 | 无法发送，提示不可为空 | 2 |
| 2.3 | 123……5000（长度大于5000） | 无法发送，提示消息长度过长 | 3 |

表11 测试结果记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例编号 | 预期结果 | 实际结果 | 测试结果 | 测试日期 | 测试人 |
| 2.1 | 发送成功 | 登陆成功 | 通过 | 2014-5-31 | 张磊 |
| 2.2 | 无法发送，提示不可为空 | 无法发送，提示不可为空 | 通过 | 2014-5-31 | 张磊 |
| 2.3 | 无法发送，提示消息长度过长 | 无法发送，提示消息长度过长 | 通过 | 2014-5-31 | 张磊 |

表12 等价类划分

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入条件 | 有效等价类 | 编号 | 无效等价类 | 编号 |
| 原密码 | 原密码与数据库密码中一致 | 1 | 原密码与数据库中密码不一致 | 4 |
| 新密码 | 长度在 6-12 之间 | 2 | 长度小于 6 | 5 |
| 长度大于 12 | 6 |
| 重复密码 | 与新密码相同 | 3 | 与新密码不同 | 7 |

表13 测试用例设计

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例编号 | 输入 | | | 预期输出 | 覆盖范围 |
| 原密码 | 新密码 | 重复密码 |
| 3.1 | 123456 | 654321 | 654321 | 修改密码成功 | 1、2、3 |
| 3.2 | 321654 | 654321 | 654321 | 修改密码失败 | 4 |
| 3.3 | 123456 | 321 | 321 | 修改密码失败 | 5 |
| 3.4 | 123456 | 123456789987 | 123456789987 | 修改密码失败 | 6 |
| 3.5 | 123456 | 654321 | 5664899 | 修改密码失败 | 7 |

注：假设数据库中包含该用户名密码为123456 的用户

表14 测试结果记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例编号 | 预期结果 | 实际结果 | 测试结果 | 测试日期 | 测试人 |
| 3.1 | 修改密码成功 | 修改密码成功 | 通过 | 2014-5-31 | 张磊 |
| 3.2 | 修改密码失败 | 修改密码失败 | 通过 | 2014-5-31 | 张磊 |
| 3.3 | 修改密码失败 | 修改密码失败 | 通过 | 2014-5-31 | 张磊 |
| 3.4 | 修改密码失败 | 修改密码失败 | 通过 | 2014-5-31 | 张磊 |
| 3.5 | 修改密码失败 | 修改密码失败 | 通过 | 2014-5-31 | 张磊 |

表15 等价类划分

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入条件 | 有效等价类 | 编号 | 无效等价类 | 编号 |
| 昵称 | 长度在 1-16 之间 | 1 | 长度为 0（即未输入） | 5 |
| 长度大于16 | 6 |
| 用户名 | 未与数据库内已有用户名重名 | 2 | 与数据库内已有用户名重名 | 7 |
| 密码 | 长度在 6-12 之间 | 3 | 长度小于 6 | 8 |
| 长度大于 12 | 9 |
| 重复密码 | 与密码相同 | 4 | 与密码不一致 | 10 |

表16 测试用例设计

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 用例编号 | 输入 | | | | 预期输出 | 覆盖范围 |
| 昵称 | 用户名 | 密码 | 重复密码 |
| 4.1 | 123 | 123 | 123456 | 123456 | 注册成功 | 1、2、3、4 |
| 4.2 | 空 | 123 | 123456 | 123456 | 注册失败 | 5 |
| 4.3 | 1234……1617（长度大于16） | 123 | 123456 | 123456 | 注册失败 | 6 |
| 4.4 | 321 | 123 | 123456 | 123456 | 注册失败 | 7 |
| 4.5 | 321 | 321 | 123 | 123 | 注册失败 | 8 |
| 4.6 | 321 | 321 | 123 | 1234……1617（长度大于16） | 注册失败 | 9 |
| 4.7 | 321 | 321 | 123456 | 654321 | 注册失败 | 10 |

表17 测试结果记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例编号 | 预期结果 | 实际结果 | 测试结果 | 测试日期 | 测试人 |
| 4.1 | 注册成功 | 注册成功 | 通过 | 2014-5-31 | 张磊 |
| 4.2 | 注册失败 | 注册失败 | 通过 | 2014-5-31 | 张磊 |
| 4.3 | 注册失败 | 注册失败 | 通过 | 2014-5-31 | 张磊 |
| 4.4 | 注册失败 | 注册失败 | 通过 | 2014-5-31 | 张磊 |
| 4.5 | 注册失败 | 注册失败 | 通过 | 2014-5-31 | 张磊 |
| 4.6 | 注册失败 | 注册失败 | 通过 | 2014-5-31 | 张磊 |
| 4.7 | 注册失败 | 注册失败 | 通过 | 2014-5-31 | 张磊 |

### 2. 对性能的测试

表18 性能测试

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **编号** | **功能名称** | **功能描述** | **输入** | **系统响应** | **发现问题** | **测试结果** | **测试人** | **测试时间** |
| 1 | 正确性需求 | 系统应能够将注册用户信息准确导入到数据库中 | 昵称、用户名、密码 | 系统将注册信息全部存入数据库 | 无 | 通过 | 张磊 | 2014-5-31 |
| 2 | 正确性需求 | 修改个人信息 | 原密码、新密码、重复密码 | 系统用现在信息替换数据库中以前的信息 | 无 | 通过 | 张磊 | 2014-5-31 |
| 3 | 正确性需求 | 系统应准确识别用户信息使用户可以正常登陆 | 用户名、密码 | 用户登陆成功 | 无 | 通过 | 张磊 | 2014-5-31 |
| 4 | 正确性需求 | 已注册的会员修改个人信息 | 用户重新输入个人信息 | 数据库中用户的信息更新 | 无 | 通过 | 张磊 | 2014-5-31 |
| 5 | 正确性需求 | 用于已注册的用户退出系统 | 点击注销按钮 | 用户退出系统 | 无 | 通过 | 张磊 | 2014-5-31 |
| 6 | 正确性需求 | 系统应能把用户发送的消息传递给对方窗口 | 输入聊天内容，点击发送按钮 | 系统将聊天内容发送到对方聊天窗口 | 无 | 通过 | 张磊 | 2014-5-31 |
| 7 | 正确性需求 | 查询相应的用户名 | 用户名 | 系统在数据库中查找相应的用户信息，返回查找到的用户信息 | 无 | 通过 | 张磊 | 2014-5-31 |
| 8 | 正确性需求 | 聊天记录查询 | 点击查询消息记录 | 系统在数据库中查找相应聊天记录，返回查询到的消息记录 | 无 | 通过 | 张磊 | 2014-5-31 |

## 六、测试结论

测试完成日期：2013-05-31

测试地点：学生公寓

测试环境：普通家用环境

参与测试的人员：张磊、王奥鑫、李小爽

系统可靠性：一般

系统强项：功能划分较为明显

系统弱项：界面不够美观

不符合项的统计结果：无

测试组组长签字：李小爽

测试组组员签字：张磊、王奥鑫